

## Profileur APMT (Gestion de fichiers)

AUTOMATE PROFILEUR MULTI-TACHES



Révision 1.0 (04.09.17)

## Sommaire

Sommaire.....	2
1. Système de fichiers.....	3
1.1 Organisation mémoire.....	3
1.2 Types de fichiers.....	3
1.3 Mémoire de travail.....	3
1.4 Mémoire de stockage permanent (carte mémoire).....	3
2. Fichiers de configuration .....	3
3. Fichiers techniques.....	4
3.1 Gestion.....	4
3.2 Données disponibles.....	5
4. Fichiers de données .....	10
4.1. Gestion.....	10
4.2. Balises.....	10
4.3. Format texte .....	11
4.4. Format binaire .....	12



## 1. Système de fichiers

### 1.1 Organisation mémoire

### 1.2 Types de fichiers

### 1.3 Mémoire de travail

### 1.4 Mémoire de stockage permanent (carte mémoire)

## 2. Fichiers de configuration

Le système est capable de générer des fichiers de configuration au format texte selon son état :

- Au démarrage de la mission
- Sur changement de configuration

Les différents fichiers générés sont nommés de la manière suivante « xxxx\_ccc\_mm\_apmt.ini » avec :

- « xxxx » : numéro de série du flotteur codé en hexadécimal (4 caractères)
- « ccc » : numéro de cycle (3 caractères)
- « mm » : numéro de motif (2 caractères)



## 3. Fichiers techniques

### 3.1 Gestion

Le système est capable de générer des fichiers techniques au format texte selon son état :

- Durant autotest
- Durant la mission
- En fin de vie

#### 3.1.1. Fichiers d'autotest

Les différents fichiers générés sont nommés de la manière suivante « xxxx\_ccc\_autotest\_nnnnn.txt » avec :

- « xxxx » : numéro de série du flotteur codé en hexadécimal (4 caractères)
- « ccc » : numéro de cycle (3 caractères)
- « nnnnn » : identification du numéro de fichier d'autotest associé au cycle (5 caractères)

#### 3.1.2. Fichiers de mission

Les différents fichiers générés sont nommés de la manière suivante « xxxx\_ccc\_mm\_technical.txt » avec :

- « xxxx » : numéro de série du flotteur codé en hexadécimal (4 caractères)
- « ccc » : numéro de cycle (3 caractères)
- « mm » : numéro de motif (2 caractères)

A noter : Un numéro de motif à « 00 » indique un fichier de pré-mission.

#### 3.1.3. Fichiers de fin de vie

Les différents fichiers générés sont nommés de la manière suivante « xxxx\_ccc\_mm\_default\_nnnnn.txt » avec :

- « xxxx » : numéro de série du flotteur codé en hexadécimal (4 caractères)
- « ccc » : numéro de cycle (3 caractères)
- « mm » : numéro de motif (2 caractères)
- « nnnnn » : identification du numéro de fichier de fin de vie associé au couple cycle/motif (5 caractères)



### 3.2 Données disponibles

Les données sont organisées dans des sections. Les différentes sections et champs les composant sont optionnels et peuvent ne pas être présents dans les fichiers.

#### [SYSTEM]

Information	Format
Offset de pression externe	Pe offset=xx.xx dbar
Pression interne	Pi=xxx.x mbar
Tension batterie à vide	Vbatt=xx.x V
Tension batterie en charge (min.)	Vbatt peak min=xx.x V
Température exteme (air)	Te air=xx.x degC
Pression externe (air)	Pe air=xx.xx dbar

#### [GPS]

Information	Format
Horodatage, position et dérive horloge	UTC=yy-mm-dd hh:mm:ss Lat=xxxx.xxxxxy Long=xxxxx.xxxxxy Clock drift=xx.xxx s

(\*) Décalage = Heure flotteur - Heure GPS. Le décalage est réinitialisé à 0 après chaque recalage GPS.

#### [IDENTIFICATION]

Information	Format
Version APMT	APMT=x.xx.xxx
Version Payload	Payload=xxxxxxxxxxxxxxx
Identification carte SIM	CID=xxxxxxxxxxxxxxx

#### [PROFILE]

Information	Format
Réduction d'urgence : - Date/heure de début - Volume huile transféré - Nb. actions électrovanne	UTC=yy-mm-dd hh:mm:ss Flotation=xxx.x cm3 (xx)
Première stabilisation - Date/heure - Profondeur	UTC=yy-mm-dd hh:mm:ss First stabilization=xxx dbar
Descente parking : - Date/heure de début - Volume huile transféré - Nb. actions électrovanne	UTC=yy-mm-dd hh:mm:ss Descent=xxx.x cm3 (xx)
Echouage : - Date/heure de début - Pression d'échouage	UTC=yy-mm-dd hh:mm:ss Grounding=xxx dbar
Echouage :	UTC=yy-mm-dd hh:mm:ss Grounding escape=xxx.x cm3



<ul style="list-style-type: none"> <li>- Date/heure de fin</li> <li>- Volume huile transféré</li> </ul>	
Dérive parking : <ul style="list-style-type: none"> <li>- Date/heure de début</li> <li>- Profondeur min.</li> <li>- Profondeur max.</li> <li>- Nb. actions électrovanne</li> <li>- Nb. actions pompe</li> <li>- Nb. rentrées consigne</li> <li>- Nb. sorties consigne</li> </ul>	UTC=yy-mm-dd hh:mm:ss Park=xxx/xxx dbar (xx/xx) stability=x/x
Dérive parking stabilisée : <ul style="list-style-type: none"> <li>- Date/heure stabilisation</li> <li>- Profondeur stabilisation</li> </ul>	UTC=yy-mm-dd hh:mm:ss Park stabilization=xxx dbar
Descente mesure : <ul style="list-style-type: none"> <li>- Date/heure de début</li> <li>- Volume huile transféré</li> <li>- Nb. actions électrovanne</li> </ul>	UTC=yy-mm-dd hh:mm:ss Deep profile=xxx.x cm3 (xx)
Dérive mesure : <ul style="list-style-type: none"> <li>- Date/heure de début</li> <li>- Profondeur min.</li> <li>- Profondeur max.</li> <li>- Nb. actions électrovanne</li> <li>- Nb. actions pompe</li> <li>- Nb. rentrées consigne</li> <li>- Nb. sorties consigne</li> </ul>	UTC=yy-mm-dd hh:mm:ss Short Park=xxx/xxx dbar (xx/xx) stability=x/x
Remontée (standard) : <ul style="list-style-type: none"> <li>- Date/heure de début</li> <li>- Volume huile transféré</li> <li>- Nb. actions pompe (total)</li> <li>- Nb. actions pour décollage</li> <li>- Profondeur maximale</li> </ul>	UTC=yy-mm-dd hh:mm:ss Ascent=xxx.x cm3 (xx/xx) from xxx dbar
Remontée (lente) : <ul style="list-style-type: none"> <li>- Date/heure de début</li> <li>- Volume huile transféré</li> <li>- Nb. actions pompe</li> </ul>	UTC=yy-mm-dd hh:mm:ss Ascent (slowly)=xxx.x cm3 (xx)
Remontée (reprise) : <ul style="list-style-type: none"> <li>- Date/heure de début</li> <li>- Volume huile transféré</li> <li>- Nb. actions pompe</li> </ul>	UTC=yy-mm-dd hh:mm:ss Ascent (resume)=xxx.x cm3 (xx)
Remontée (fin) <ul style="list-style-type: none"> <li>- Date/heure de fin</li> </ul>	UTC=yy-mm-dd hh:mm:ss Ascent end
Surface <ul style="list-style-type: none"> <li>- Date/heure de début</li> </ul>	UTC=yy-mm-dd hh:mm:ss Surface
Accrochage : <ul style="list-style-type: none"> <li>- Date/heure de début</li> <li>- Pression d'accrochage</li> </ul>	UTC=yy-mm-dd hh:mm:ss Hanging=xxx dbar
Accrochage : <ul style="list-style-type: none"> <li>- Date/heure de fin</li> </ul>	UTC=yy-mm-dd hh:mm:ss Hanging escape
Dérive parking (glace) <ul style="list-style-type: none"> <li>- Date/heure de début</li> <li>- Profondeur min.</li> </ul>	UTC=yy-mm-dd hh:mm:ss Ice Park=xxx/xxx dbar (xx/xx) stability=x/x



<ul style="list-style-type: none"> <li>- Profondeur max.</li> <li>- Nb. actions électrovanne</li> <li>- Nb. actions pompe</li> <li>- Nb. rentrées consigne</li> <li>- Nb. sorties consigne</li> </ul>	
Dérive parking stabilisée (glace) : <ul style="list-style-type: none"> <li>- Date/heure stabilisation</li> <li>- Profondeur stabilisation</li> </ul>	UTC=yy-mm-dd hh:mm:ss Ice Park stabilization=xxx dbar
Activation pressostat	UTC=yy-mm-dd hh:mm:ss Pressure switch activation
Remontée d'urgence <ul style="list-style-type: none"> <li>- Date/heure de début</li> </ul>	UTC=yy-mm-dd hh:mm:ss Emergency ascent

**[DATA]**

Information	Format
Emission données (*) : <ul style="list-style-type: none"> <li>- Taille totale données</li> <li>- Nb. fichiers</li> <li>- Débit moyen</li> <li>- Nb. sessions</li> </ul>	Upload=xx.x kB of x file(s) at x.x kB/min in x session(s)
Réception télécommande : <ul style="list-style-type: none"> <li>- Nb. acceptées</li> <li>- Nb. refusées</li> <li>- Nb. inconnues</li> </ul>	Download=command file (x accepted, x refused, x unknown)
Réception configuration Payload	Download=payload file
Réception script	Download=script file
Nombre de fichiers associés au motif (**)	Pattern=x files
Nombre de points par capteur : <ul style="list-style-type: none"> <li>- Nom capteur (SBE41, ...)</li> <li>- Nb. descente parking</li> <li>- Nb. dérive parking</li> <li>- Nb. descente mesure</li> <li>- Nb. dérive mesure</li> <li>- Nb. remontée</li> <li>- Nb. sub-surface</li> </ul>	xxxxx=x/x/x/x/x/x points

(\*) Informations du motif/cycle précédent.

(\*\*) Ne sont pas pris en compte les fichiers : technique d'autotest ou de pré-mission, de configuration ni d'acquittement de commandes/scripts. Le nombre de fichiers est exprimé avant découpage des fichiers pour transmission.

**[POWER]**

Information	Format
Durée du motif	Pattern=x min
Rapport traitement/veille	Treatment=xx %
Cumul activations hydrauliques : <ul style="list-style-type: none"> <li>- Electrovanne</li> <li>- Pompe</li> </ul>	EV/Pump=xxx/xxx cs





Cumul activation SBE41	SBE41=xxx min
Cumul activation modem (*)	Transmission=xxx min
Cumul activation GPS	GPS=xxx s

(\*) Informations du motif/cycle précédent.

### [ALARM]

Information Déploiement	Format
Mise en route	Power-on
Configuration invalide	Bad configuration
Flotteur trop lourd	Flotation (heavy)
Flotteur trop léger	Flotation (light)
Echec autotest : - Source(s) xxx parmi « FRAM, FLASH, RTC, Vbatt, Pi, Pe, SBE41-Cutoff, SBE41-Offset, GPS, Payload, Sensor(xxx), Transmitter »	Autotest fail=xxx, ...
Etat	
Tension batterie à vide faible	Vbatt low
Tension batterie en charge faible	Vbatt peak low
Pression externe faible	Pe low (xxx dbar)
Pression externe forte	Pe high (xxx dbar)
Capteur de pression externe	Pe broken
Saut de vitesse	Pe SR high
Pression interne forte	Pi high
Présence d'eau	Water inside
Navigation	
Atteinte du fond à la descente	Grounding (xxx dbar)
Accrochage à la remontée	Hanging (xxx dbar)
Freinage à la descente	Braking
Erreurs de fonctionnement	
Défaut système	System
Défaut carte Payload	Payload
Défaut GPS	GPS
Défaut hydraulique	Hydraulic
Défaut ADC	ADC
Défaut de fichier	File (skip)
Défaut RTC	RTC
Défaut pressostat	Pressure switch
Changement de mode	
Fin de vie : - Source xxx parmi « Pi high, Pe broken, Pe high, Vbatt low, Vbatt peak low,	End of life (xxx)





Water inside, Flotation (heavy), Flotation (light) »	
Procédure de survie	Rescue
Rétroactions : <ul style="list-style-type: none"> <li>- Type xxx parmi « Early profile, Early surface, Abort profile, Abort cycle 0, Abort cycle 1, Go to deep, Standard speed, Lower speed”</li> <li>- Nb. acceptées</li> <li>- Nb. refusées</li> </ul>	Feedback=xxx (x accepted, x refused) Feedback=xxx (x accepted, x refused) ...

**Exemple :****[SYSTEM]**

Pi=750.3 mbar

Vbatt=10.7 V

Vbatt peak min=10.2 V

**[PROFILE]**

UTC=15-09-03 16:32:26 Flotation=6.6 cm3 (2) down to 17 dbar

UTC=15-09-03 16:33:09 Descent=7.5 cm3 (5)

UTC=15-09-03 17:28:26 Park=51/51 dbar (0/0) stability=0/0

UTC=15-09-03 17:32:29 Deep profile=0.0 cm3 (0)

UTC=15-09-03 17:35:11 Ascent=15.2 cm3 (3/2) from 54 dbar

UTC=15-09-03 17:47:29 Ascent end

**[POWER]**

Pattern=89 min

Treatment=2 %

EV/Pump=221/448 cs

SBE41=79 min

**[ALARM]**

Payload



## 4. Fichiers de données

### 4.1. Gestion

Le système est capable de générer des fichiers de données capteurs selon trois formats :

- Format texte (\*.csv)
- Format binaire standard (\*.hex)
- Format binaire étendu (\*.hex)

Les différents fichiers de mesure générés sont nommés de la manière suivante

« xxxx\_ccc\_mm\_ssssss.ext » avec :

- « xxxx » : numéro de série du flotteur codé en hexadécimal (4 caractères)
- « ccc » : numéro de cycle (3 caractères)
- « mm » : numéro de motif (2 caractères)
- « sssss » : identification du capteur (N caractères, exemple : « sbe41 »)
- « ext » : extension du fichier

### 4.2. Balises

Tous les fichiers sont organisés sous forme de sections grâce à des balises d'identification de la phase et du type de traitement.

#### 4.2.1. Identification de la phase de navigation

A chaque changement de phase de navigation, une chaîne d'identification est insérée. Les différentes chaînes ASCII sont les suivantes :

- **[DESCENT]** : Données issues de la phase de descente depuis la surface jusqu'à la profondeur de dérive
- **[PARK]** : Données issues de la phase de dérive
- **[DEEP\_PROFILE]** : Données issues de la phase de descente depuis la profondeur de dérive jusqu'à la profondeur de mesure
- **[SHORT\_PARK]** : Données issues de la phase de dérive à la profondeur de mesure
- **[ASCENT]** : Données issues de la phase de remontée

#### 4.2.2. Identification du traitement

A chaque changement de phase ou zone de traitement, une chaîne d'identification des traitements est insérée. Les différentes chaînes ASCII sont les suivantes :

- **(RW)** : Pour données brutes
- **(AM)** : Pour moyenne arithmétique
- **(SD)** : Pour écart type
- **(MD)** : Pour médiane
- **(SS)** : Pour point de sub-surface



Plusieurs traitements peuvent se suivre. Les combinaisons sont les suivantes :

- (RW)
- (AM)
- (AM)(SD)
- (AM)(MD)
- (AM)(SD)(MD)
- (SS)

### 4.3.Format texte

Format tableur avec possibilité de choisir le séparateur de décimal, le séparateur de colonnes ainsi que le format d'horodatage.

[DESCENT]									
(AM) (SD) (MD)									
2015-06-15	08:34:51	4.90	17.4640	35.798	0.0000	0.0000	4.90	17.4640	35.798
2015-06-15	08:34:51	5.50	17.4600	35.797	0.0030	0.0000	5.50	17.4600	35.797
2015-06-15	08:34:51	6.40	17.4530	35.797	0.0030	0.0000	6.40	17.4530	35.797
2015-06-15	08:34:51	7.40	17.4470	35.796	0.0030	0.0000	7.40	17.4500	35.797
2015-06-15	08:34:51	8.50	17.4380	35.796	0.0030	0.0000	8.50	17.4360	35.796
2015-06-15	08:34:51	9.50	17.4320	35.796	0.0030	0.0000	9.50	17.4290	35.796



## 4.4.Format binaire

### 4.4.1. Format de codage

Les données sont enregistrées au format « little endian » :

- Octet (char)
- Entier (2 octets –short int)
- Entier long (4 octets –long int)
- Horodatage (4 octets –long int) - format Unix Epoch – 1<sup>er</sup> janvier 1970 à 0h00
- Flottant (4 octets - float)

### 4.4.2. Structure des fichiers

Chaque fichier commence par un octet d'identification du codage :

- 0x01 : Format étendu
- 0x02 : Format standard (compact)

Ensuite, suivent des groupes d'enregistrements identifiables grâce aux balises.

**Exemple** : Fichier format standard, phase de descente

00000000:	02 5b 44 45 53 43 45 4e 54 5d 28 41 4d 29 28 53	. [DESCENT] (AM) (S
00000010:	44 29 28 4d 44 29 2b 6c be a0 31 00 c0 57 d6 8b	D) (MD) +1% 1.ÀWÖ<
00000020:	00 00 31 00 c0 57 d6 8b 37 00 bc 57 d5 8b 03 00	..1.ÀWÖ< 7.4WÖ< ..
00000030:	37 00 bc 57 d5 8b 40 00 b5 57 d5 8b 03 00 40 00	7.4WÖ< @.µWÖ< ..@.
00000040:	b5 57 d5 8b 4a 00 af 57 d4 8b 03 00 4a 00 b2 57	µWÖ< J. WÖ< ..J. *W
00000050:	d5 8b 55 00 a6 57 d4 8b 03 00 55 00 a4 57 d4 8b	Ö< U. ;WÖ< ..U. *WÖ<
00000060:	5f 00 a0 57 d4 8b 03 00 5f 00 9d 57 d4 8b 69 00	_ . WÖ< .._ .WÖ< i.
00000070:	99 57 d3 8b 03 00 69 00 95 57 d2 8b 73 00 92 57	WÖ< ..i. *WÖ< s. 'W
00000080:	d2 8b 05 00 73 00 92 57 d2 8b 7e 00 87 57 d1 8b	Ö< ..s. 'WÖ< ~. #WÑ<
00000090:	00 00 7e 00 87 57 d1 8b 85 00 87 57 d1 8b 00 00	..~. #WÑ< .... #WÑ< ..
000000a0:	85 00 87 57 d1 8b 90 00 7d 57 d1 8b 05 00 90 00	.... #WÑ< . }WÑ< ...
000000b0:	7d 57 d1 8b 9a 00 79 57 d0 8b 00 00 9a 00 79 57	}WÑ< š.yWÖ< ..š.yW

**A noter** : En transmission Iridium RUDICS, les fichiers peuvent être complétés par des octets de « padding » (0x1A). Ces octets ne doivent pas être décodés. Chaque fichier peut se terminer par une suite de 0 à 1023 octets de « padding ».

### 4.4.3. Codage des données

Les données sont codées de la manière suivante :

- Pression en cbar (entier non signé)
- Température en m°C et offset de +5.0°C (entier non signé)
- Salinité en mpsu (entier non signé)



- Ecart type température en m°C (entier signé)
- Ecart type salinité en mpsu (entier signé)

#### 4.4.4. Format binaire standard (compact)

Ce format correspond à la version actuellement en place sur les autres flotteurs nke :

- Horodatage du premier point de chaque phase et/ou zone de traitement
- Précision standard

Durant les phases de navigation « descente » et « remontée » le premier point de chaque zone de traitement est horodaté en absolu, puis les enregistrements de points suivants ne sont par horodatés.

Durant les phases de « dérive » tous les points sont horodatés en absolu.

Les structures d'enregistrements sont les suivantes :

##### Sub-surface

```
struct
{
    unsigned long int uliDateTime;
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
}
tSSStd;
```

##### Données brutes (dérive)

```
struct
{
    unsigned long int uliDateTime;
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
}
tRStdPark;
```

##### Données brutes (navigation)

```
struct
{
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
}
```



```
tRStdNav;
```

### Données moyennées

```
struct
{
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
}
tMStd;
```

### Données moyennées et écart-type

```
struct
{
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
    signed char cStdDeviationTemperature;
    signed char cStdDeviationSalinity;
}
tMECStd;
```

### Données moyennées et médiane

```
struct
{
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
    unsigned short int uiMedianPressure;
    unsigned short int uiMedianTemperature;
    unsigned short int uiMedianSalinity;
}
tMMStd;
```

### Données moyennées, écart-type et médiane

```
struct
{
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
    signed char cStdDeviationTemperature;
    signed char cStdDeviationSalinity;
    unsigned short int uiMedianPressure;
    unsigned short int uiMedianTemperature;
}
```



```

unsigned short int uiMedianSalinity;
}
tMECMStd;

```

### Exemple : Enregistrement

2015-06-15 08:34:51

4.9;17.464;35.798;0.000;0.000;4.9;17.464;35.798

5.5;17.460;35.797;0.003;0.000;5.5;17.460;35.797

00000000:	02	5b	44	45	53	43	45	4e	54	5d	28	41	4d	29	28	53
00000010:	44	29	28	4d	44	29	2b	6c	be	a0	31	00	c0	57	d6	8b
00000020:	00	00	31	00	c0	57	d6	8b	37	00	bc	57	d5	8b	03	00
00000030:	37	00	bc	57	d5	8b	40	00	b5	57	d5	8b	03	00	40	00
00000040:	b5	57	d5	8b	4a	00	af	57	d4	8b	03	00	4a	00	b2	57
00000050:	d5	8b	55	00	a6	57	d4	8b	03	00	55	00	a4	57	d4	8b
00000060:	5f	00	a0	57	d4	8b	03	00	5f	00	9d	57	d4	8b	69	00
00000070:	99	57	d3	8b	03	00	69	00	95	57	d2	8b	73	00	92	57
00000080:	d2	8b	05	00	73	00	92	57	d2	8b	7e	00	87	57	d1	8b
00000090:	00	00	7e	00	87	57	d1	8b	85	00	87	57	d1	8b	00	00
000000a0:	85	00	87	57	d1	8b	90	00	7d	57	d1	8b	05	00	90	00
000000b0:	7d	57	d1	8b	9a	00	79	57	d0	8b	00	00	9a	00	79	57

. [DESCENT] (AM) (S  
D) (MD) +1% 1.ÀWÖ<  
..1.ÀWÖ< 7.ÀWÖ< ..  
7.ÀWÖ< @.µWÖ< ..@.  
µWÖ< J. WÖ< ..J. \*W  
Ö< U. !WÖ< ..U. \*WÖ<  
\_. WÖ< ..\_. WÖ< i.  
WÖ< ..i. \*WÖ< s. 'W  
Ö< ..s. 'WÖ< ~. #WÑ<  
..~. #WÑ< .... #WÑ< ..  
... #WÑ< . }WÑ< ...  
}WÑ< š. yWÖ< ..š. yW

Pression = 49 cbar (0x0031)





#### 4.4.5. Format binaire étendu

L'intérêt de ce format par rapport à la version compact est :

- L'horodatage de chaque point
- L'amélioration de la résolution des enregistrements (pression et température)

Durant les phases de navigation « descente » et « remontée » le premier point de chaque zone de traitement est horodaté en absolu (date de référence), puis les enregistrements de points suivants sont horodatés en codant en relatif la date par rapport à la date de référence (écart en secondes).

Durant les phases de « dérive » tous les points sont horodatés en absolu.

L'extension de résolution des mesures de pression et température est codée sur un octet partagé en utilisant des masques, de la manière suivante :

- Extension pression (0,01 dbar) = valeur & 0xF0
- Extension température (0,1 m°C) = valeur & 0x0F

Les structures d'enregistrements sont les suivantes :

##### Sub-surface

```
struct
{
    unsigned long int uliDateTime;
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
    unsigned char ucAveragePTExtra;
}
tSSExt;
```

##### Données brutes (parking)

```
struct
{
    unsigned long int uliDateTime;
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
    unsigned char ucAveragePTExtra;
}
tRExtPark;
```

##### Données brutes (navigation)



```

struct
{
    unsigned short int uiDateTimeDelta;
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
    unsigned char ucAveragePTExtra;
}
tRExtNav;

```

### Données moyennées

```

struct
{
    unsigned short int uiDateTimeDelta;
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
    unsigned char ucAveragePTExtra;
}
tMExt;

```

### Données moyennées et écart-type

```

struct
{
    unsigned short int uiDateTimeDelta;
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
    unsigned char ucAveragePTExtra;
    signed char cStdDeviationTemperature;
    signed char cStdDeviationSalinity;
}
tMECExt;

```

### Données moyennées et médiane

```

struct
{
    unsigned short int uiDateTimeDelta;
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
    unsigned char ucAveragePTExtra;
    unsigned short int uiMedianPressure;
    unsigned short int uiMedianTemperature;
    unsigned short int uiMedianSalinity;
}

```



```

    unsigned char ucMedianPTExtra;
}
tMMExt;

```

### Données moyennées, écart-type et médiane

```

struct
{
    unsigned short int uiDateTimeDelta;
    unsigned short int uiAveragePressure;
    unsigned short int uiAverageTemperature;
    unsigned short int uiAverageSalinity;
    unsigned char ucAveragePTExtra;
    signed char cStdDeviationTemperature;
    signed char cStdDeviationSalinity;
    unsigned short int uiMedianPressure;
    unsigned short int uiMedianTemperature;
    unsigned short int uiMedianSalinity;
    unsigned char ucMedianPTExtra;
}
tMECMEExt;

```

### Exemple : Enregistrement

2015-06-15 10:13:00 **5.53**; 17.4606; 35.797; 0.003; 0.000; 5.55; 17.4574; 35.797  
 2015-06-15 10:14:45; 6.53; 17.4535; 35.797; 0.003; 0.000; 6.52; 17.4503; 35.797

00000000:	01	5b	44	45	53	43	45	4e	54	5d	28	41	4d	29	28	53	. [DESCENT] (AM) (S
00000010:	44	29	28	4d	44	29	2c	83	be	a0	00	00	37	00	bc	57	D) (MD), f% ..7.4W
00000020:	d5	8b	36	03	00	37	00	b9	57	d5	8b	54	69	00	41	00	Ô< 6...7.2WÔ< Ti.A.
00000030:	b5	57	d5	8b	35	03	00	41	00	b2	57	d5	8b	23	dc	00	μWÔ< 5...A.*WÔ< #Û.
00000040:	4a	00	ae	57	d4	8b	77	03	00	4a	00	ae	57	d4	8b	57	J.©WÔ< w...J.©WÔ< W
00000050:	be	00	55	00	a6	57	d4	8b	88	03	00	55	00	a4	57	d4	%U.¡WÔ< ^...U.*WÔ
00000060:	8b	91	6e	00	5f	00	a0	57	d4	8b	42	03	00	5f	00	9d	< 'n._. WÔ< B..._.
00000070:	57	d4	8b	10	04	01	69	00	99	57	d3	8b	01	03	00	69	WÔ< ...i.™WÔ< ...i
00000080:	00	95	57	d2	8b	09	be	00	73	00	92	57	d2	8b	83	05	.*WÔ< .%.s.'WÔ< f.
00000090:	00	73	00	92	57	d2	8b	83	0f	00	7e	00	87	57	d1	8b	.s.'WÔ< f...~.#WÑ<
000000a0:	57	00	00	7e	00	87	57	d1	8b	57	0a	00	85	00	87	57	W...~.#WÑ< W.....#W
000000b0:	d1	8b	97	00	00	85	00	87	57	d1	8b	97	0f	00	90	00	Ñ< -.....#WÑ< -...

Pression = 55 cbar (0x0037) + 0,03 dbar (0x3x) = 5,53 dbar



## nke Instrumentation

Rue Gutenberg, ZI de Kerandré  
56700 Hennebont, France

Tel (+33) 2 97 36 10 12 – Fax (+33) 2 97 55 17

[www.nke-instrumentation.com](http://www.nke-instrumentation.com)

